Arjen.Lenstra@epfl.ch, Professeur au Laboratoire de cryptologie algorithmique, EPFL-IC

Les standards en cryptographie sont-ils souhaitables?

Les décideurs ne doivent pas se préoccuper de cryptologie tant qu'ils se conforment aux standards. Mais ils doivent se sentir un peu plus concernés quand de nouveaux standards sont annoncés, avec des propriétés incertaines, inattendues ou indésirables. Nous allons présenter ici de récents développements dans cette thématique.

La cryptographie, l'art et la science de l'écriture secrète, est au cœur technique de la Sécurité de l'Information. Les outils de base de cryptographie qui sont présents partout sont:

- les fonctions de hachage, pour obtenir rapidement *l'empreinte* de documents;
- le chiffrement symétrique, pour des chiffrement et déchiffrement rapides de grandes quantités de données;
- l'échange de clé, pour négocier la clé utilisée lors de chiffrement symétrique;
- les signatures électroniques, pour signer les hachages cryptographiques des documents.

Les choix sont incroyablement multiples pour chacun de ces outils (par contraste avec le peu d'outils mathématiques impliqués – voir l'article **Les problèmes mathématiques à la base de la sécurité de l'information** de ce numéro). Cela ne concerne pas les responsables de la sécurité informatique des institutions, ceux qui sont en charge de faire les bons choix pour les technologies et les produits. La seule chose dont ils doivent se soucier c'est de suivre les bonnes pratiques tout en se conformant aux standards industriels existants ou virtuels, et le cas échéant, aux règles. Comme ces bonnes pratiques le sont réellement et comme les standards sont le résultat d'années d'études de la part des experts du monde entier, il semble qu'il n'y ait pas de problème à se conformer à l'approche habituelle de mise en œuvre de la sécurité de l'information.

Néanmoins, c'est loin d'être aussi évident. Il est généralement admis que le niveau voulu de sécurité des outils standard cryptographiques n'est plus suffisant (simplement parce que

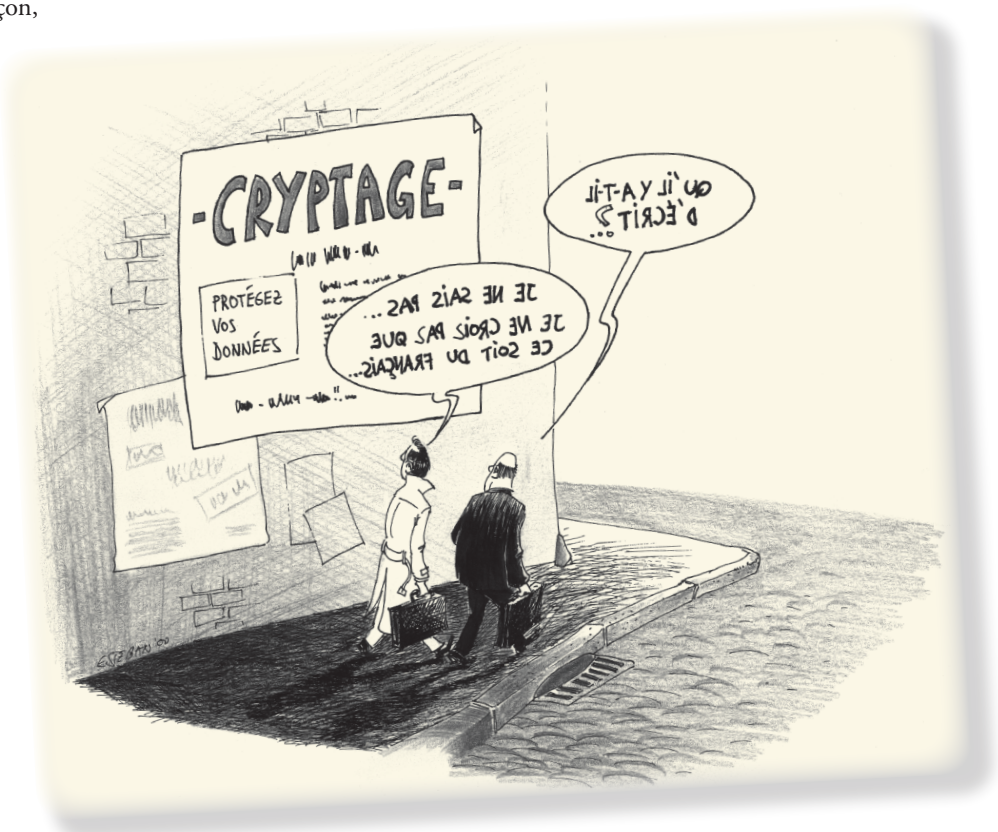
les processeurs deviennent de plus en plus rapides). Pour noircir encore un peu plus le tableau, on a démontré qu'un des outils standard –et largement utilisé- n'atteint pas ce niveau voulu de sécurité.

C'est pourquoi, de nouveaux standards sont en train de naître: selon les pages Web du NIST des États-Unis (*National Institute of Standards and Technology*)[1], ***Suite B Cryptography*** va montrer dans quelle direction aller pour atteindre un niveau satisfaisant de sécurité cryptographique vers 2010. C'est la NSA *(US National Security Agency)*[2] qui a récemment annoncé *Suite B Cryptography* qui est censé fournir à l'industrie un ensemble d'algorithmes cryptographiques qui pourront être utilisés pour créer des produits répondant à la majorité des besoins des agences gouvernementales états-uniennes. Bien que visant les besoins du gouvernement US, on ne peut douter que l'utilisation de *Suite B Cryptography*, ou du moins de certaines parties, se répandra pour une utilisation plus générale. En conséquence, l'approche conformiste habituelle entraînera rapidement les sociétés du monde entier à adopter les méthodes de *Suite B Cryptography.* Elles auraient pu le faire de toute façon, mais faire partie de la *Suite B Cryptography* implique un tampon d'approbation qui permet d'éviter les critiques qui, sinon, n'auraient pas manqué de s'élever.

Suite B est sur le principe une initiative méritoire. Elle soulève aussi quelques questions, la première par le but exprimé. Sur le site Web de Suite B, on peut lire **Les avancées constantes et rapides des technologies de l'information au 21e siècle réclament l'adoption d'une stratégie cryptographique flexible et souple pour protéger l'information concernant la sécurité nationale**. Au vu des événements récents – particulièrement le fait que des méthodes cryptographiques bien établies et largement utilisées se sont avérées avoir des propriétés indésirables inattendues- il est en effet obligatoire d'adopter une approche *flexible et souple* dans l'utilisation de méthodes cryptographiques.

Le moins que l'on puisse dire c'est qu'il est déconcertant que cette flexibilité et cette souplesse n'apparaissent pas dans *Suite B Cryptography:* en fait, on ne fait qu'y proposer une seule méthode cryptographique pour chacun des quatre outils de base cités plus haut. La possibilité de remplacer rapidement un outil *cassé*, ce qui est une sage précaution etant donné les progrès de la cryptanalyse, n'est pas une exigence pour la *Suite B Cryptography*. La page Web du NIST autorise un peu plus de flexibilité pour le choix de la signature électronique et les outils d'échange de clés, mais ne met pas non plus assez de poids sur le besoin de souplesse.

Le manque de souplesse peut être encore empiré par le choix de quelques outils cryptographiques actuels. Voici la situation: le seul outil de hachage de *Suite B Cryptography* est SHA-2, qui se rapporte à un certain nombre de fonctions de hachage cryptographique avec différents niveaux de sécurité cryptographique. Aujourd'hui, il n'a pas de raison de soupçonner que les hachages de SHA-2 n'atteignent pas les niveaux de sécurité prévus. Mais il y a malgré tout un souci. En terme de conception les hachages de SHA-2 sont les derniers membres d'une famille de hachage qui comprend MD4, MD5, SHA-0 et SHA-1 (dans l'ordre chronologique). MD4, et partiellement MD5 et SHA-0, étaient déjà connus pour être plus fragiles que prévu. En 2004 et 2005, on savait que tous ces *vieux* hachages avaient de sérieux problèmes de conception. Les faiblesses qui en découlent sont difficiles à exploiter et on peut argumenter que beaucoup d'applications ne sont pas concernées. Par ailleurs, SHA-2 ne semble pas partager ces problèmes avec ses prédécesseurs. Néanmoins, l'émergence de problèmes sous-jacents indique que l'état de l'art dans la conception des fonctions de hachage en cryptographie fait gravement défaut. Apparemment l'expérience et le savoir qui ont été injectés dans la conception de fonctions comme MD5 et SHA-1 étaient insuffisants. Comparé à SHA-1, SHA-2 contient plusieurs modifications supposées intelligentes qui jusqu'à présent semblent fonctionner, car les cryptanalyses actuelles de



[1] *csrc.nist.gov/ispab/2006-03/E_Barker-March2006-ISPAB.pdf*

[2] *www.nsa.gov/ia/industry/crypto_suite_b.cfm*

SHA-1 n'affectent pas SHA-2. Mais pour l'unique fonction de hachage cryptographique incluse dans un nouveau standard, on aimerait disposer d'une base plus solide pour avoir vraiment confiance. Malheureusement, il n'y a pas d'alternative universellement acceptée à SHA-2. Ne serait-il pas préférable de reporter la migration coûteuse vers Suite B jusqu'à ce qu'on dispose d'une meilleure alternative?

Le seul outil de chiffrement symétrique inclus dans la *Suite B Cryptography* est le célèbre AES *(Advanced Encryption Standard)*. C'est sans aucun doute un algorithme bien conçu, résultat d'une compétition internationale et choisi en 2000 par NIST après un soigneux processus de sélection. Depuis son invention à la fin des années 90, aucune faille sévère n'a été découverte dans la méthode de chiffrement AES malgré de sérieuses attaques par des cryptanalystes de par le monde. Néanmoins, un problème récent nous fait nous demander si on aurait choisi la même méthode s'il avait été connu en 2000.

Ce problème est que les implémentations sous forme logicielle de AES sont particulièrement vulnérables aux attaques dites *par le cache*[3]. L'attaquant observe simplement le comportement et le *timing* d'un de ses *process* (pas directement lié à AES) qui tourne en même temps et sur le même processeur que le *process* du logiciel AES attaqué. Il est montré que les données rassemblées peuvent conduire à des informations concernant la clé secrète AES, qui peut ensuite être connue en une fraction de seconde. En général, on croit que les privilèges d'accès sont suffisants pour offrir une protection et une séparation adéquates entre des *process* partageant le même processeur. Cette attaque, cependant, ne réclame aucun privilège par rapport au process AES, il suffit d'avoir un accès simultané au processeur sur lequel le logiciel AES s'exécute.

Pendant la *compétition* AES, le fait que le cache soit une ressource partagée qui permet des fuites d'information entre les process était soit ignoré soit considéré hors de propos. Actuellement, tout logiciel incluant un process AES est à risque s'il s'exécute sur un processeur qui peut être partagé. C'est le cas des serveurs et de tout ordinateur sans un contrôle d'accès approprié, ce qui est à peu près le cas de la majorité des machines connectées à Internet. Des mesures de protection ont été proposées pour faire échouer les attaques par le cache, mais elles ne sont pas fréquemment appliquées et ne font pas partie de l'approche standard de la sécurité informatique ni de ses bonnes pratiques. On peut noter que plusieurs fonctions de chiffrement autres que AES sont aussi vulnérables aux attaques par le cache, mais aussi que certains des candidats finalistes pendant la compétition AES sont censés être moins vulnérables.

Une fois ce nouveau fait connu, quid de AES et de son inclusion dans la *Suite B Cryptography?* On aimerait y voir des conseils sur les circonstances dans lesquelles AES ne doit pas être utilisé ou sur quel type de précautions on doit prendre. Peut-être que ces précautions sont des pratiques standard dans certains environnements, mais dans la plupart des cas elles ne le sont pas. Restreindre l'utilisation d'AES uniquement à des implémentations *hardware* n'est pas une option réaliste. Il n'est pas plus réaliste de restreindre l'usage d'AES à des environnements non partagés –c'est tout le contraire qui se passe, avec l'intérêt croissant de l'industrie pour les machines virtuelles, on partage de plus en plus les ressources de calcul, avec la conséquence évidente d'un risque d'attaques par le cache.

Pour les deux derniers outils, l'échange de clé et les signatures électroniques, *Suite B Cryptography* utilise la cryptographie par courbes elliptiques (*Elliptic Curve Cryptography* - ECC) et suggère l'usage d'un certain nombre de courbes elliptiques standard. Tant qu'il n'y a pas de *pairings* (qui ne sont d'ailleurs pas présents dans la suite B), ECC est une technique mature que l'on peut à présent recommander pour une application pratique. Cependant, se reposer exclusivement[4] sur une technologie qui peut exiger une licence[5] pour une utilisation en dehors du gouvernement états-unien peut faire sourciller certains. En outre, l'utilisation de courbes standard avec des nombres premiers spécialement formatés, est encore une autre contradiction par rapport à *une stratégie cryptographique flexible et souple*.

Pour conclure ce court article, que nous apporte le conformisme en cryptographie? C'est en général le pari le plus sûr. Mais il n'est pas évident qu'une conformité aveugle à la *Suite B Cryptography* soit la meilleure stratégie. Il serait bienvenu que se mette en place une discussion plus large sur la situation actuelle, en particulier sur la confiance de la *Suite B* uniquement dans SHA-2 comme hachage cryptographique et dans AES comme outil de chiffrement symétrique, sur la nécessité de licence pour ECC, et sur son étonnant manque de souplesse et de flexibilité.

Une remarque -ou pensée- d'une tout autre nature: a-t-on vraiment besoin d'un plus haut niveau de sécurité cryptographique? Dans la plupart des circonstances industrielles pratiques, les niveaux de sécurité globaux ne s'approchent même pas du niveau de sécurité cryptographique bientôt inadéquat fourni par les standards cryptographiques actuels. Améliorer la cryptographie ne fait rien dans ce sens. On peut se demander quel est le vrai retour sur investissement[6].

(traduction de l'original anglais par Jacqueline.Dousson@epfl.ch, Domaine IT)

[3] Eprint.iacr.org/2005/271

[4] Le site Web du NIST laisse la porte ouverte à RSA et aux cryptosystèmes traditionnels basés sur les logarithmes discrets, avec des clés de grande taille.

[5] Selon ***www.nsa.gov/ia/industry/crypto_suite_b.cfm***: *tout vendeur construisant des produits pour un usage touchant à la sécurité nationale doit recevoir une licence de la part de NSA.*

[6] Mes remerciements à Benne de Weger et Paul Hoffman pour leurs commentaires.